DOCUMENT RESUME

ED 421 140 IR 018 860

AUTHOR Comer, Priscilla Garrido; Geissler, Colin
TITLE A Methodology for Software Evaluation.
PUB DATE 1998-00-00
NOTE 7p.; In: "SITE 98: Society for Information Technology & Teacher Education International Conference (9th, Washington, DC, March 10-14, 1998). Proceedings"; see IR 018 794.
PUB TYPE Guides - Non-Classroom (055) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Check Lists; Computer Assisted Instruction; *Computer Software Evaluation; *Computer Software Selection; *Courseware; Educational Environment; Educational Objectives; Educational Technology; Evaluation Criteria; *Evaluation Methods; Foreign Countries; *Instructional Material Evaluation; Learning Theories

ABSTRACT

Evaluators of software for education must make a series of decisions about which issues have a direct impact on their choice of software. Instructional context analysis is the first step, including identifying the learners, the instructor, the learning environment, and technical needs and limitations. The next step is instructional goal analysis; in order to define this statement, the evaluator needs to consider views of software (i.e., various taxonomies of software) and learning theory. The following are issues to address when developing and evaluating a personalized software checklist: (1) content, including quality, depth, and tests; (2) interface, including ease of use, navigation, text quality, graphics, and sound; (3) interactivity, including feedback, sequence, and questions; (4) classroom-related issues, including entry level/technical requirements, motivation, backwash, and management; and (5) support, including online and off-line help. (Contains 32 references.) (AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 421 140

A Methodology for Software Evaluation

By:

Priscilla Garrido Comer
Colin Geissler

IR019860

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

G.H. Marks

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

A Methodology for Software Evaluation

Priscilla Garrido Comer
Concordia University

Colin Geissler
Concordia University

When attempting to evaluate software for education, the goals of the learners and the instructors need to be explicitly stated from the outset. For example, considering a specific piece of software adequate simply because it is labeled 'multimedia' is not acceptable. As pointed out by Nyns (1989), a teacher's starting point in using Computer Assisted Language Learning (CALL) should not be "what can I do with my PC?" but rather "which medium is best suited to teach such and such a skill?" (p. 36). To which we would add - to these students, in this school, and on this type of computer system.

Evaluators must make a series of decisions about which issues have a direct impact on their choice of software. Instead of attempting to find an existing checklist that is in agreement with their own purposes, they should create their own checklists based on what they have determined to be the pertinent issues.

The procedure is as follows: evaluators define their instructional context by determining (a) who the learners are; (b) who the instructor is and the learning environment; and (c) what the 'raw' technical limitations are. Consequently, with this defined, the instructional goal may be determined. This, in turn, helps them develop statements about the type of software that will meet their needs. At this point, they would choose and weigh checklist items which they believe will aid in answering the question: "What is the best software in this context?". Potential software candidates can then be identified and evaluated using this newly created, context-specific checklist.

Instructional Context Analysis

If one wishes to make a prudent purchase, one must analyze, beforehand, the context in which materials will be used — to ensure that various needs are met. A new instructional technique may only be appropriate in certain situations, as is the case with computer software. The time spent conducting this analysis will save time and money, as the products which are purchased will better reflect the needs of those affected.

Identify Learners

Smith and Ragan (1993) have produced an outline of learner characteristics that can be used to analyze and describe almost any target audience. The list is quite lengthy (for explanations and examples see the original text) and it is suggested that one not use every category. Instead, it should suffice to use only those which are most relevant to the context and which help in delimiting the characteristics and interests of the given learner population.

a. Cognitive Characteristics. (i.e. Development level (such as Piaget's levels of cognitive development), Reading level)
b. Psychosocial Characteristics. (i.e. Interests, Motivations, Anxiety level)
c. Physiological Characteristics. (i.e. Sensory perception (visual and auditory acuity), Age)

Identify Instructor and the Learning environment

This step is important to ensure that if a purchase is made, the product will be given a positive reception and some environmental support is in place to secure its effective use in the classroom (Smith & Ragan, 1993, p. 31).

Issues that are important to examine because they directly address the needs of the teacher and therefore inform the software evaluation are:

- Feelings towards computers in education - What are the educator's attitudes and opinions? If teachers are reluctant or negative then the mere introduction of software — regardless of its quality — may be a waste of time and money. Administrative support or training may be helpful.
- Interests and preferences of teacher - What are her interests outside the content matter? Some software may be more intrinsically motivating due to its relevance to her interests and teaching styles.
- Experience level of teacher - How much experience does she have with the content, learners, media and teaching in general?
- Curricular/instructional goals - What are the specific goals that should be addressed by the product?

- Administrative needs - Will the teacher need the software to collect and save data for administrative purposes such as attendance; achievement; time on task; and the paths the learner has taken through the software?
- Administrative environment - Do the administrators support the integration of computer technology in the classroom? Can they be counted upon to provide training, incentives and/or time-off for the instructors to become proficient with computers?

Identify Technical Needs

This may be the most simple and straightforward aspect of the instructional context analysis (depending upon the technical knowledge of the evaluator). The purpose of this stage is to explicitly state the maximum and minimum requirements of the physical resources on which software will be used.

- Who? The best sources for the information one requires are the people who purchased, installed and maintain the system. The issues to consider when speaking to these informants are:
- What? Type of computer - Determine operating system; processor speed; size of hard drive, RAM; monitor type (color?) and size; multimedia capabilities (sound, video and CD-ROM speed); modem speed; and networking capabilities.
- What else? Peripheral devices - What other kinds of devices are attached to the computers? Printers, scanners, mice, microphones, removable media drives or projection systems.
- Where? Number and configuration - How many computers are there? Are they in a lab or in classrooms? Are they connected by a network and if so, what kind? Do they have Internet access?

Instructional Goal Analysis

"Ask not what a computer can do to students, but what students can do with a computer." David Thornberg (as quoted in Ryan, 1997)

The next step is to determine the instructional goal that the software should fulfill. This goal will give context to the ultimate question "What is the best software?". It will contain the pertinent information about the learner; teacher and learning environment; and technical needs. It will also include a statement related to the functionality and/or type of software desired; and the instructor's beliefs about the learning process.

The terms which help define this statement depend upon some philosophical issues that the evaluator needs to consider:

View of software

Software that will be used in the classroom comes in many forms. Taylor (1980) created a distinction among software - tutor, tool and tutee (e.g. Romeiser & Yerem, 1994). A tutor can be seen as a surrogate teacher, a tool takes student input and generates output which is used by the student in a task, and software that students use to 'teach' the computer are considered tutees. Software that is not instructional in nature would not fit into these categories. For example, Taylor's taxonomy (1980) cannot properly deal with Internet browsers which are not instructional but have the potential to become standard applications in many classrooms.

More importantly, it may be argued that these classifications look at the question of computers in education only from a technological point of view (what can the computer do?) and hardly consider what many would feel is the most integral part of the instructional context - the learner (what can the student do with the computer?).

Using Dewey's (1943) 'Natural impulses of the learner', Bruce & Levin (in press) suggest that software be classified by how the learner can use it in meeting these impulses — thus the categories media for inquiry, media for communication, media for construction and media for expression emerge. This change from the 'technocentric' to a more learner centered point of view fundamentally changes how one phrases the statement of the instructional goal.

For example, if one asks "What kinds of games teach punctuation in an English as a Second Language (ESL) classroom?", an assumption has already been made that a game — a category that is becoming more difficult to define — is the appropriate option. The exploration of possible software titles has been needlessly limited. However, if one looks from the learner's perspective, the question becomes something like: "With what media can a student examine (media for inquiry) and practice punctuation in an ESL classroom?". This question, in turn, may be answered with a punctuation game, concordance application, grammar checker and the traditional 'paper and pencil' workbook. The scope of possible applications has been greatly expanded beyond those that the category 'game' could yield.

Theory of Learning

Software developers have beliefs about how people learn and interact in different environments. By critically examining and comparing computer applications, often these beliefs can be deduced. In the same way, teachers have strong beliefs about how best certain content or concepts can be taught to different learners.

If an evaluator wants to find appropriate software that will be used to fulfill a particular instructional goal, then the learning theory which was used in the program's development must reflect the teacher's tacit theories of learning and instruction.

As a consequence, during this process, "software may be found inappropriate rather than ineffective" (Miller &

Burnett, 1986, p. 159). With reference to the previous example, if the teacher believes strongly that students can best examine punctuation inductively and then practice it by writing compositions then a 'behavioristic' game would not be appropriate and probably would never be used in that classroom.

Therefore, at this stage the evaluator would ask the question "What does the teacher believe is the best way for the instructional goal to be met?" and then ensure that possible software choices reflect this. With this information, the evaluator will have once again narrowed the focus of the search and made the selection process more efficient and, possibly, more successful.

Issues to consider when developing a personalized software evaluation checklist

Content
- Quality
  - Learners
    - Appropriate vocabulary and subject matter for target audience
    - Well written and grammatically correct texts
    - Variety of explanations and examples
  - Teacher
    - Features which take advantage of the computer
    - Easy procedure for modifying, adding or updating content
    - Clearly stated objectives
    - Outlines, summaries and reviews
- Depth
  - Learner
    - Versatility
    - Numerous and 'just in time' explanations, examples, and illustrations
    - Presents 'correct' examples, non-examples and mistakes most commonly made by learners.
    - Situates the examples within a context
  - Teacher
    - Applicable in a variety of teaching/learning situations
    - Meets the instructional goals
- Tests
  - Teacher
    - Systematic review of learned materials
    - Coincide with objectives
    - Presented appropriately

Interface
- Ease of Use
  - Learner
    - Entered answers or selected alternatives should be able to be deleted.
    - Changes to the environment through menu options
  - Teacher
    - The program is simple to use
  - Technical
    - Simplicity in its loading and running instructions.
- Navigation
  - Technical
    - Timing is self-paced and flexible
    - Ways to go back one step or go back to the main menu, etc.
    - Help or hint options
    - Exit and review instructions
    - Text or images that are hyperlinks should be clearly indicated
    - The 'granularity' (number of links to other screens) should be level appropriate
    - The cursor should not appear if it is not possible to use it
- Text Quality
  - Learner
    - Font is large enough
    - Clear and easy to read (including text inside buttons)
    - Glossed, with pop-up references
    - Symbols and icons should be obvious or, if tried once, easy to remember
- Graphics
  - Learner
    - Style and graphics are suitable
    - Graphics clarify or enhance the points being made
    - Familiar objects should be included in illustrations
    - Illustrations are clearly labeled and as close as possible to the text to which they refer
  - Technical
    - Graphics would improve the program
    - Still images do not interfere with the text
    - Consistent use of symbols
    - Overall quality and clarity of displays
    - "Current video standard (QuickTime)
- Sound
  - Learner
    - Voices are intelligible
  - Technical
    - Sound constitutes an essential or integral part of the program
    - Appropriate use of voice over music
    - Quality recording and playback features

Interactivity
- Feedback
  - Learner
    - Helpful messages to aid in the correction of errors.
    - Types of feedback
    - Immediate
    - Delayed
    - Summative
    - Types of branching feedback
    - Review
    - Remedial Audio
    - Single/Multiple Remediation
    - Unpredictable patterns

Technical
- Processing time

Sequence
Learner
- Linear or fixed sequence
- User directed
- Hyperlinks
- 'Open' in that there is no predetermined sequences
- Avoids putting learners in infinite loops or sending them to dead ends

Teacher
- The instructor can order the topics for presentation

Technical
- Acceptable time delays

Questions
Learner
- Those asked by learner
- Questions of clarification
- Key word searches
- Complete sentences
- Those asked of learner
- Categories of questions
- Appropriate to the audience
- Related to the content
- Randomized

Classroom related issues
Entry Level / Technical Requirements
Learner & Teacher
- Entry level skills and knowledge (keyboarding, mouse, navigation)
- Amount of basic or remedial training required
- Level of computer literacy skills

Technical
- Compatibility of the system's requirements and the minimum requirements
- The availability of site and distribution licenses
- Total cost

Motivation
Learner
- Motivating and challenging as a whole
- Novelty (vs. lasting) effect

Backwash
Teacher
- Provides teacher with information/insight into students' strengths and weaknesses
- Provides teacher with ideas for instruction
- Structure/content mirrors classroom activities
- Allows teacher to 'offload' some tasks

Management
Learner
- Creates a 'history' of the path taken through the program
- Allows one to save one's work/position
- Multiple-user profiles

Teacher
- Ease of implementation
- Audit Trails
- Diagnosis of difficulties

Support
On-Line Help
Student
- Key word / topic searching
- Glossary / Index
- Pop-up menus
- Tutorials
- Ability to annotate

Teacher
- Ability to annotate student help

Technical
- Unambiguous on-screen options
- Uses 'help' key
- Other on-line help

Off-Line Help
Student
- Pre-Printed workbooks and guides
- Print-outs of help topics and/or annotations
- Lists of suggestions for study, resources or bibliographies

Teacher
- Clear instructions on how to operate the program
- Instructor's guide

Technical
- Clarity in its documentation
- Telephone support and fax-back services

References

Alessi, S. M., & Trollip, S. R. (1991). *Computer-based instruction: Methods and development*. Englewood Cliffs, NJ: Prentice-Hall.

Blease, D. (1983) Observer effects on teachers and pupils in classroom research. *Educational Research*, 35(3), p. 213-217.

Blease, D. (1986). *Evaluating Educational Software*. London: Croom Helm.

Bloom, B. (1956). Taxonomy of educational objectives: cognitive domain. David MacKay. In Brown, H. (1993). *Teaching by principles*. New Jersey: Prentice Hall Regents.

Borsook, T. (1991). Harnessing the power of interactivity for instruction in Schwier, R. & Misanchuk, E. (1993). *Interactive Multimedia Instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Brown, H. (1993). *Teaching by principles*. New Jersey: Prentice Hall Regents.

Brown, H. (1994). *Principles of language learning and teaching*. New Jersey: Prentice Hall Regents.

Bruce, B. & Levin, J. (in press). Educational Technology: Media for Inquiry, Communication, Construction, and Expression. *Journal of Educational Computing Research*.

Cohen, V. (1983). Criteria for the Evaluation of Microcomputer Courseware. *Educational Technology*, Jan. 1983, pp. 9-14 in Cunningham, P. (1995). Evaluating CALLware for your classroom. *CAELL Journal*, 6(3), 13-19.

Council of Ministers of Education of Canada. (1985). *Software evaluation : criteria for educational computer software evaluation*. Toronto : Council of Ministers of Education of Canada.

Cunningham, P. (1995). Evaluating CALLware for your classroom. *CAELL Journal*, 6(3) 13-19.

Dewey, J. (1943). *The school and society*. Second edition. Chicago: University of Chicago Press.

Gallier-Morgan, H. *McBOOKmaster* [On-line]. Available: http://www.uqtr.uquebec.ca/eif/mcmaster.html#McBOOKmaster

Golub, L. (1983). English and language education courseware inventory. *Focus: Teaching English Language Arts*, 9, 117-123.

Hannafin, M. (1989). Interaction strategies and emerging instructional technologies: Psychological perspectives. *Canadian Journal of Educational Communications*, 18(3), 167-179. In Schwier, R. & Misanchuk, E. (1993). *Interactive Multimedia Instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Healey, D. What makes software worth buying? *CAELL Journal*, 5(1), 12-15.

Hughes, A. (1989). *Testing for Language Teachers*. New York: Cambridge University Press.

Jay, T. (1983). The cognitive approach to computer software courseware design and evaluation. *Educational Technology*, Jan. 1983, pp. 22-26 in Cunningham, P. (1995). Evaluating CALLware for your classroom. *CAELL Journal*, 6(3), 13-19.

Kinsella, K. (1991). Promoting active learning and classroom interaction through effective questioning strategies. Workshop presented at San Francisco State University, September 14. In Brown, H. (1993). *Teaching by principles*. New Jersey: Prentice Hall Regents.

Locatis, C., & Atkinson, F. (1984) *Media and Technology for Education and Training*. Ohio: Charles E. Merrill Publishing Company.

Nyns, R. (1989). Is intelligent computer assisted language learning possible? *System*, 17, 35-37.

Palmberg, R. (1981). CALL Checklist in *MEUSLI News*, March 1994, pp. 5-7.

Porras, H. (1993). *Design and formative evaluation of a prototype hypermedia program on development of hypermedia*. Masters Thesis. Montreal: Concordia University.

Ragsdale, R. (1982). *Evaluation of micro computer courseware. Computers in Education Series*. Toronto: OISE Press.

Romeiser, & Yerem (1994). Learn to speak english for the multimedia PC, Ver 3.0. *TESL-EJ* [On-line]. Available: http://www.kyoto-su.ac.jp/information/tesl-ej/ej01/mr.1.html

Ryan, K. (1997). *Multimedia Software and Language Learning* [On-line]. Available: http://www.swu.ac.jp/~ryan/rmenu.html

Schramm (1964). The research on programmed instruction: An annotated bibliography. Washington: US Department of Health, Education and Welfare, Office of Education. Publication no. OE 34034 in Smith, P. & Ragan, T. (1993). *Instructional Design*. New York, NY: Macmillan Publishing company.

Schwier, R., & Misanchuk, E. (1993). *Interactive Multimedia Instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Smith, P., & Ragan, T. (1993). *Instructional Design*. New York, NY: Macmillan Publishing company.

Taylor, R. (1980). *The computer in the school: tutor, tool, tutee*. New York: Teachers College Press.

Wiener, N. (1954). *The human use of human behaviour*. New York: Houghton Mifflin Co.

Zamel, V., Cybernetics: A model for feedback in the ESL classroom. *TESOL Quarterly*, 15(2), 139-150.

Priscilla Garrido Comer is a Master of Arts in Educational Technology candidate at Concordia University, Montreal, Quebec, Canada and a media specialist at New Horizons Bilingual School, Juan Miguel Roman II, Bella Vista, Santo Domingo, Dominican Republic. E-mail: p.garrido@codetel.net.do

Colin Geissler is a Master of Arts in Educational Technology candidate at Concordia University, Montreal, Quebec, Canada. E-mail: ah917@freenet.carleton.ca